

Please type a plus sign (+) inside this box ⇒



201-345-00

PTO/SB/05 (12/97)

Approved for use through 09/30/00 OM 0651-0032

Patent and Trademark Office U S DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

02/25/98
1c572 U.S. PTO

UTILITY PATENT APPLICATION TRANSMITTAL <small>(Only for new nonprovisional applications under 37 CFR 1.53(b))</small>	Attorney Docket No.	M-3206-1C US	Total Pages (This form)	2
	First Named Inventor or Application Identifier			
	Peter C. Chen			
	Express Mail Label No.	EM 263 040 735 US		

APPLICATION ELEMENTS <small>See MPEP chapter 600 concerning utility patent application contents</small>	ADDRESS TO: Assistant Commissioner for Patents Box Patent Application Washington, D C 20231
1. <input checked="" type="checkbox"/> Fee Transmittal Form - <i>see page 2 of this form.</i> <small>(Submit an original, and a duplicate for fee processing)</small> 2. <input checked="" type="checkbox"/> Specification [Total Pages (Total 20 pages)] <small>(preferred arrangement set forth below)</small> <ul style="list-style-type: none"> • Descriptive title of the Invention • Cross References to Related Applications • Statement regarding Fed sponsored R&D • Reference to Microfiche Appendix • Background of the Invention • Brief Summary of the Invention • Brief Description of the Drawings (if filed) • Detailed Description • Claim(s) • Abstract of the Disclosure 3. <input checked="" type="checkbox"/> Drawing(s) (35 USC 113) <small>[Total Sheets 5]</small> 4. <input checked="" type="checkbox"/> Oath or Declaration <small>[Total Pages 2]</small> <ul style="list-style-type: none"> a. <input type="checkbox"/> Newly executed (original or copy) b. <input checked="" type="checkbox"/> Copy from a prior application (37 CFR 1.63(d)) <small>(for continuation/divisional with Box 17 completed)</small> c. <input type="checkbox"/> <u>DELETION OF INVENTOR(S)</u> <small>Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b)</small> 5. <input checked="" type="checkbox"/> Incorporation By Reference <small>(useable if Box 4b is checked)</small> The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.	6. <input checked="" type="checkbox"/> Microfiche Computer Program (Appendix) 7. Nucleotide and/or Amino Acid Sequence Submission (if applicable, all necessary) <ul style="list-style-type: none"> a. <input type="checkbox"/> Computer Readable Copy b. <input type="checkbox"/> Paper Copy (identical to computer copy) c. <input type="checkbox"/> Statement verifying identity of above copies
ACCOMPANYING APPLICATION PARTS	
8. <input type="checkbox"/> Assignment Papers (cover sheet & documents) 9. <input type="checkbox"/> 37 CFR 3.73(b) Statement <input type="checkbox"/> Power of Attorney 10. <input type="checkbox"/> English Translation Document (if applicable) 11. <input type="checkbox"/> Information Disclosure Statement (IDS)/PTO-1449 <input type="checkbox"/> <u> </u> Copies of IDS Citations 12. <input checked="" type="checkbox"/> Preliminary Amendment 13. <input checked="" type="checkbox"/> Return Receipt Postcard (MPEP 503) <small>(should be specifically itemized)</small> 14. <input checked="" type="checkbox"/> Small Entity Statement <input checked="" type="checkbox"/> Statement filed in prior application; status still proper and desired <input type="checkbox"/> Is no longer claimed. 15. <input type="checkbox"/> Certified Copy of Priority Document(s) <small>(if foreign priority is claimed)</small> 16. <input checked="" type="checkbox"/> Other: Copy of Special Petition for Extension of Time	

17. If a CONTINUING APPLICATION, check appropriate box and supply the requisite information:
☒ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No. **08/428,935, (Filed 4/25/95)**

18. CORRESPONDENCE ADDRESS

☐ Customer Number or Bar Code Label or ☐ Correspondence address below

Name	David T. Millers			Reg. No. 37,396	
Attorneys for Applicant	SKJERVEN, MORRILL, MacPHERSON, FRANKLIN & FRIEL LLP				
Address	25 Metro Drive, Suite 700				
City	San Jose	State	CA	Zip Code	95110
Country:	United States	Telephone	(408) 453-9200	Fax	(408) 453-7979



19. Fee calculations.

CLAIMS (Number Filed)	(1) FOR	(2)		(3) NUMBER EXTRA		(4) RATE		(5) CALCULATIONS
10	TOTAL CLAIMS (37 CFR 1.16(c))	-20	=	0	x	\$22	=	\$ 0.00
2	INDEPENDENT CLAIMS (37 CFR 1.16(b))	-3	=	0	x	\$82	=	\$ 0.00
	MULTIPLE DEPENDENT CLAIMS (if applicable) (37 CFR 1.18(d))				+	\$270.00	=	\$ 0.00
						BASIC FEE (37 CFR 1.16(a))	=	\$790.00
						Total of above Calculations	=	\$ 790.00
						Reduction by 50% for filing by small entity (Note 31 CFR 1.9, 1.27, 1.28).	=	\$ 395.00
						TOTAL	=	\$ 395.00

20. **FEES:** The Commissioner is hereby authorized to credit overpayments or charge the following fees to Deposit Account No. **19-2386**:

- a. ☒ Fees required under 37 CFR 1.16. (U.S. Application Filing Fees)
 b. ☒ Fees required under 37 CFR 1.17. (Conditional Extension of Time Fees)
 c. ☐ Fees required under 37 CFR 1.18. (Patent Issue Fees)

21. ☒ Other: _____

NOTE:

The prior application's correspondence address will carry over to this UPA UNLESS a new correspondence address is provided below.

22. NEW CORRESPONDENCE ADDRESS

☐ Customer Number or Bar Code Label

☐ New correspondence address below

NAME			
ADDRESS			
CITY	STATE	ZIP CODE	
COUNTRY	TELEPHONE	FAX	

23. SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

Skjerven, Morrill, MacPherson, Franklin & Friel LLP

25 Metro Drive, Suite 700

San Jose, CA 95110

Tel. (408) 453-9200

Fax. (408) 453-7979

Date: February 25, 1998

Name: David T. Millers

Reg. No. 37,396

Signature

David Millers

Express Mail
Label No.

EM 263 040 735 US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Peter C. Chen
Assignee: PC-Tel, Inc.
Title: COMMUNICATIONS INTERFACE AND CONFLICT AVOIDANCE
USING A SOFTWARE SIMULATION OF A UART
Serial No.: Unknown Filed: February 25, 1998
Examiner: Unknown Group Art Unit: Unknown
Docket No.: M-3206-1C US

San Jose, California
February 25, 1998

COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D. C. 20231

PRELIMINARY AMENDMENT

Sir:

Please amend the accompanying application as follows.

IN THE SPECIFICATION

Page 1, line 5, insert --This patent document is a continuation U.S. patent application Ser. No. 08/428,935, entitled "Host Signal Processing Modem Using a Software Simulation of a UART," filed April 25, 1995.--

Page 2, line 20, delete "on".

Page 7, line 3, before "Appendix", insert --Microfiche--.

Page 7, line 26, change "provides" to --provided--.

Page 8, line 23, after "driver", insert --can--.

Page 13, line 5, change "register are" to --registers contain--.

Page 13, line 6, change "bit" to --bits--.

IN THE CLAIMS

1. (Amended) A system comprising:
a computer having a processing unit, a main memory, and a local bus;

LAW OFFICES OF
SKJERVEN, MORRILL,
MacPHERSON, FRANKLIN
& FRIEL LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453 9200
FAX (408) 453-7979

408784 v1

a device coupled to the local bus, wherein the device occupies an I/O slot [which corresponds to] and is accessible at a first [communications port on the local bus] set of address, and the device has a register set [which] with an address assignment in the first set of addresses that differs from a standard address assignment of a register set for a UART corresponding to the I/O slot;

an operating system executed by the processing unit, wherein the operating system includes a procedure for accessing a register set of a UART corresponding to the first communications port; and

a communications driver executed by the processing unit, the communication driver comprising [:] a UART emulation which in response to [execution of] an access by the procedure for accessing [a] the register set of a UART, [accesses storage locations in the main memory] converts the access as required for the register set and address assignment of the device [; and an I/O handler which transfers values between the storage locations in the main memory and the register set of the device].

4. (Amended) A method for communication between a computer and a device having an I/O interface which differs from the I/O interface of a UART, comprising:

coupling the I/O interface of the device to a local bus in the computer;

allocating in a memory of the computer, storage locations which correspond to registers of a UART; and

transmitting [values] information via the local bus between the I/O interface of the device and the allocated storage locations in the memory of the computer.

5. (Amended) The method of claim 4, further comprising transmitting from an application to a communications driver a [data] packet [which is] formatted for a UART, wherein the communication driver updates a value in the storage locations according to a value in the [data] packet.

6. (Amended) The method of claim 5, wherein the communication [drive] driver performs the step of transmitting by:

converting a value [in the data packet] from the allocated storage to a converted value compatible with the I/O interface of the device; and

writing the converted value to a register in the device via the local bus.

7. (Amended) The method of claim 4, wherein transmitting further [comprising] comprises:

reading values from a register of the device via the local bus; and
updating the storage locations according to the value read.

8. (Amended) The method of claim 7, further comprising transmitting from a communications driver to an application [a data packet containing a value] information from the storage locations.

10. (Amended) The method of claim 9, wherein setting the base device address comprises:

sensing, by the device, of a data signal on the local bus;
comparing [by] the data signal to a signal from a pattern generator in the device;
advancing a state indicator toward a final state in response to the data signal being equal to the signal from the pattern generator;
repeating the steps of sensing, comparing, and advancing until the state indicator reaches the final state; and
setting the base address of the device to a value indicated by a signal on the local bus in response to the state indicator reaching the final state.

Please cancel claims 11-16.

REMARKS

The accompanying application is a continuation of U.S. patent application Ser. No. 08/428,935. The above amendment amends claim 1 to a form not previously presented in the parent application. Claims 4-8 and 10 are amended to the form those claims had in the parent application when canceled in the parent application. Claims 11-16 are canceled.

Applicant requests entry of the above amendment before examination of the accompanying application. The Examiner is invited to call the undersigned at

(408) 453-9200 if there are any questions concerning the application or this preliminary amendment.

EM 263 040 735 US

Respectfully submitted,

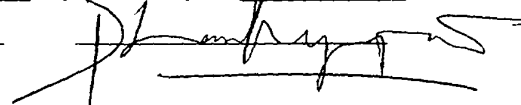
David Millers

David T. Millers
Attorney for Applicant(s)
Reg. No. 37,396

2025 OCT 14 09:00

I hereby certify that this correspondence is being deposited with the United States Postal Service as express mail in an envelope addressed to: Commissioner of Patents and Trademarks, Washington, D.C., 20231, on April 25, 1995; Express Mail Receipt No. TB 856 096 22X US.

H-25-95
Date of Signature



COMMUNICATIONS INTERFACE AND CONFLICT AVOIDANCE

USING A SOFTWARE SIMULATION OF A UART

Peter C. Chen

5

REFERENCE TO MICROFICHE APPENDIX

The present specification comprises a microfiche appendix A. The total number of microfiche sheets in the microfiche appendix is one. The total number of frames in the microfiche appendix is fifteen.

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever.

20

BACKGROUND OF THE INVENTION

Field of the Invention

This invention relates to a computer system including a device having a non-standard I/O interface coupled to a local bus and a software emulation of a universal asynchronous receiver transmitter (UART), and further relates to processes and circuit for avoiding conflicts when assigning a COM port to a non-standard device.

30

Description of Related Art

A typical personal computer (PC) has one or more a local buses such as ISA, VESA, and/or PCI buses for connection of user-selected devices. The PC communicates with the devices using device addresses

055220 07205000

typically indicated by the settings of jumper wires or toggle switches on the device. Problems can arise because there is no guaranty that a set of devices, made by different manufacturers, can operate together without address conflicts. Even if a set of devices can operated together, connection of the devices to a local bus may require that the user identify address conflicts and change address settings to avoid the conflicts. This can make adding devices to a PC difficult.

Common devices connected to an ISA bus include serial input/output (I/O) devices such as a printer, a modem, or a mouse. Some operating environments such as Microsoft WINDOWS™ running in conjunction with MS-DOS operating system provide for standardized connections to serial devices coupled to the ISA bus. In particular, WINDOWS™ and MS-DOS support four communication or COM ports, each having a predefined base device address. This allows resolution of device address conflicts if each serial device on coupled to the ISA bus has settings for at least four different base device addresses. Each COM port is for connection to a serial device which contains a communication interface known as a Universal Asynchronous Receiver/Transceiver (UART). The UART is well known in the art and described, for example, in the 1994 "Telecommunication Data Book" from National Semiconductor Corporation, which is incorporated by reference herein in its entirety.

Fig. 1 illustrates conventional communications between a serial device 110 and an application 140 via an operating environment 130 and a communications driver 120. Operating environment 130 provides a library of subroutines which application 140 calls to communicate with serial device 110. The subroutines call communications driver 120 which writes and reads

data and control information to and from a UART 105.
The standardized communication interface illustrated in
Fig. 1 reduces the complexity of application 140
because application 140 is not required to implement a
5 variety of communication protocols. Accordingly, most
application's are written for the standard interface.
However, a standard hardware UART may be unsuitable or
too expensive for some devices. Accordingly,
techniques are needed which provide non-standard
10 devices with the benefits of a standard UART interface.

SUMMARY OF THE INVENTION

09030710-022598
In accordance with the invention, a software
emulation of a UART (universal asynchronous receiver
15 transmitter) allows a device with a non-standard I/O
interface to communicate with an application through an
operating environment which contains procedures for
accessing standard UART interfaces. The software UART
allows a non-standard device to take advantage of
20 protocols which avoid device address conflicts among
COM ports. Further, differences between the non-
standard device and a standard UART device are
transparent to the applications running under the
operating environment.

25 In one embodiment of the invention, a computer
system includes a non-standard device and a COM driver
for the non-standard device. The non-standard device
connects to an I/O slot corresponding to a first COM
port but has a register set which differs from the
30 standard register set for a UART. The COM driver
contains: a UART emulation which in response a
procedure requesting access to a register of a UART at
the first COM port, instead accesses storage locations
in main memory of the computer system; and an I/O
35 handler which transfers values between the storage
locations in main memory and the register set of the

device. Optionally, the system includes a standard device having a UART coupled to an I/O slot corresponding to a second COM port, and the COM driver contains routines for accessing the standard device.

5 To avoid address conflicts with standard devices, the non-standard device has a circuit for setting the device address of the non-standard device. In one embodiment, this circuit contains a comparator adapted for receiving a data signal from the local bus and for
10 comparing the data signal to a pattern signal which has a predetermined series of values; a counter coupled to the comparator, wherein the counter resets to an initial state if the comparator indicates the data signal is not equal to the pattern signal and advances
15 toward a final state if the comparator indicates the data signal equals the pattern signal; and a register which, in response to the counter reaching the final state, latches from the local bus a value which indicates the base address of the non-standard device.
20 Typically, the circuit also contains an address decoder that selects which data signals the comparator receives from the local bus.

365200 DT 200000
25 The COM driver sets the base address of the non-standard device by sending a predetermined pattern of address and data signals on the local bus and then following the pattern with a signal that indicates the base address of the device. The device starts in a locked state where the device does not have a base address and does not respond to signals on the local
30 bus. Once the device recognizes the pattern sent by the COM driver, the device address is set to the value provided by the COM driver, and the device transitions to an unlocked state. In the unlocked state, the device responds to signals on the local bus which
35 correspond to the base address of the device.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows a block diagram of a prior art interface between an application and a UART device.

Fig. 2 shows a block diagram of an interface in accordance with an embodiment of the invention.

Fig. 3A is a block diagram of a circuit which unlocks a device and sets the base device address of the device.

Fig. 3B is a block diagram of an embodiment of a base address decoder usable in the circuit of Fig. 3A.

Fig. 3C is a block diagram of an embodiment of a pattern generator usable in the circuit of Fig. 3A.

Use of the same reference symbols in different figures indicates similar or identical items.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

An embodiment of this invention illustrated in Fig. 2 allows an application 140 running in an operating environment 130 to communicate with a serial device 110 having a hardware UART 105 and/or a serial device 210 having non-standard input/output (I/O) interface 205. A COM driver 220 contains conventional software subroutines for communications with hardware UART 105 and a software UART 222 for serial device 210. Software UART 222 allows operating environment 130 and application 140 to transparently communicate with non-standard serial device 210 as if serial device 210 contained a hardware UART.

In one embodiment of the invention operating environment 130 includes microsoft WINDOWSTM which supports four COM ports for communications with up to four serial devices connected to an ISA bus 115. A standard COM port occupies a slot of eight addresses on ISA bus 115. The eight addresses correspond to registers of a standard UART, which function as shown in Table 1.

00000710-022598

Table 1

	Offset	Register
	0	Rx/Tx Buffer (read/write) or Divisor Latch least significant byte.
	1	Interrupt Enable or Divisor Latch most significant byte.
5	2	Interrupt Identification
	3	Line control
	4	Modem control
	5	Line status
	6	Modem status
10	7	Scratch

The divisor latch indicated in Table 1 is enabled by setting a bit DLAB in the line control register.

Serial device 210 logically occupies a COM port but does not have a hardware UART which physically occupies an I/O address slot on ISA bus 115. Accordingly, the I/O slot for the COM port used by serial device 210 is available for non-standard interface 205. Non-standard I/O 205 occupies up to eight addresses on ISA bus 115 but need not comply with the standard functions given in Table 1. An example non-standard interface is disclosed below.

Application 140 can be any sort of software. A typical application 140 is a communication program that transmits and receives data through a modem. To access a device connected to ISA bus 115, application 140 calls a routine in operating environment 130. The routine calls COM driver 220, and COM driver 220 accesses devices 110 and/or 210 via ISA bus 115. COM driver 220 is software containing a standard COM driver for UART 105 and a software UART 222 and an I/O handler 224 for communications with non-standard I/O device 210. A computer running application 140 executes

software of COM driver 220 when operating environment 130 calls COM driver 120 and during interrupts.

Appendix A contains a listing of 8086 assembly language program which implements software UART 222 and I/O handler 224. Software UART 222 contains a set of virtual registers which are memory locations in the computer running COM driver 220 and which correspond to the registers of a standard UART. The virtual registers are updated using information from serial device 210 and operating environment 130. I/O handler 224 accesses serial device 210 (hardware) which is referred to as the ASIC in Appendix A.

During initialization of COM ports, COM driver 220 determines which of the four COM ports are allocated to standard UART devices, determines if a non-standard device is present, and then allocates an unassigned COM port and I/O slot to device 210. Serial device 210 is initially locked during start-up. When locked, serial device 210 receives a data signal DATA and address signal ADDR from ISA bus 115, but does not respond to any address. COM driver 220 unlocks device 210 by transmitting address signal ADDR and data signal DATA with values equal to predefined pattern recognized by device 210. When unlocked, the base device address of device 210 depends on information that COM driver 220 provides while unlocking device 210. Device 210 replies to the address set by COM driver 220 to indicate that device 210 is present.

Fig. 3A shows a block diagram of an unlocking circuit 300 which unlocks a device coupled to a local bus of a computer. Although, unlocking circuit 300 is described herein in the context of a serial device coupled to an ISA bus, unlocking circuit 300 is more generally applicable to any device coupled to a local bus such as a VLB or PCI bus.

Unlocking circuit 300 contains a base address decoder 330 and a pattern generator 310. While the device is locked, base address decoder 330 asserts a signal SEL to pattern generator 310. Pattern generator 310 generates a signal PAT that represents a byte which is from a predefined sequence and corresponds to the value of signal SEL. Signal SEL starts in an initial state, such as indicating a count value of zero or a maximum count. Each time the local bus carries an address signal ADDR having a recognized value, base address decoder 330 compares data signal DATA from the local bus to signal PAT and if signals PAT and DATA are equal, changes signal SEL so that signal SEL advances toward a final state. Otherwise, signal SEL is reset to indicate to the initial state. Advancing signal SEL can for example increment a count value from an initial state (minimum value) toward a final state (maximum value) or decrement the count from an initial state (maximum value) to a final state (minimum value).

When signal SEL reaches the final state, base address decoder 330 receives and stores a base address for the device and then asserts a signal PCSYNC to indicate the device is unlocked. The COM driver transmit the base address to the device in a number of ways. For example, the address signal used during transmission of the pattern or a following data signal can indicate the base address.

Fig. 3B shows a block diagram of an embodiment of base address decoder 330. Base address decoder 330 contains AND gates 331, 332, and 333 which are coupled address lines of ISA bus 115. AND gate 333 asserts a signal ADX when signal IOWCN indicates the computer is writing data and address signal ADDR[11:0] has the form 001x 111x 1111 binary where x indicates that bits ADDR4 and ADDR8 are "don't care" bits, i.e. can have either value 0 or 1. The COM driver selects values for bits

ADDR8 and ADDR4 so that signal ADDR[11:0] does not correspond to any other device coupled to the ISA bus. A signal AEN indicates when a DMA controller in the computer places an address on ISA bus 115. In the embodiment shown, unlocking circuit 300 does not respond to the DMA controller, and signal ADX is only asserted if signal AEN indicates address signal ADDR [11:0] is not from the DMA controller.

AND gate 333 deasserts signal ADX when at the end of a write cycle and causes register 339 to latch a byte from signal DATA[7:0] on ISA bus 115. A comparator 340 compares the latched byte to signal PAT[7:0] and asserts a signal EQUAL if the latched byte equals the byte indicated by signal PAT[7:0]. Signal EQUAL determines what occurs the next time signal ADX is asserted. Signal EQUAL acts as a clock enable signal for a counter 335 and a flip-flop 345 and acts as an input signal for flip-flop 341. With signal EQUAL asserted when AND gate 333 asserts signal ADX, counter 335 increments count signal SEL[2:0], flip-flop 341 asserts a signal MATCH, and flip-flop 345 sets signal PCSYNC to the value of bit SEL2 of signal SEL[2:0]. With signal EQUAL deasserted when AND gate 333 asserts signal ADX, flip-flop 341 deasserts signal MATCH which resets counter 335 to an initial state with count value zero. Thus, counter 335 is reset to zero each time a data byte from ISA bus 115 is not equal to the data byte from the predefined pattern. Additionally, a signal RESET from ISA bus 115 can reset counter 335.

Count signal SEL[2:0] increments if signal EQUAL is asserted when COM driver 220 generates on ISA bus 115 an address of the form 001x 111x 1111 and a data byte equal to signal PAT[7:0]. Incrementing signal SEL[2:0] causes pattern generator 310 to set signal

PAT[7:0] to indicate the next byte in the predefined pattern.

Fig. 3C is a block diagram of an embodiment of pattern generator 310. Pattern generator 310 contains multiplexers 311 to 318 which have select terminals coupled to receive signal SEL[2:0]. Input terminals of multiplexers 311 to 314 are coupled to voltage VCC or ground. Signal SEL0 selects one of the two values for a signal AOUT[7:0] from multiplexers 311 and 312 and one of two values for a signal BOUT[7:0] from multiplexers 313 and 314. Multiplexers 315 and 316 select an output signal DOUT[7:0] which is equal to either signal AOUT[7:0] or BOUT[7:0] depending on select signal SEL1. Multiplexers 317 and 318 select signal PAT[7:0] which is equal to signal DOUT[7:0] or a fixed value depending on the value of select signal SEL2. Pattern generator 310 generates a five byte string, the ASCII code for "Pctel", which indicates the manufacturer of the device.

Many alternative patterns and pattern generators may be used in place of the embodiment shown in Fig. 3C. For example, pattern generator 310 can be implemented using a memory such as a read-only memory where signal SEL[2:0] is an address signal or implemented using combinatorial logic where signal SEL[2:0] is an input signal. Each value in the pattern can be longer or shorter than a byte and can be a constant value independent of signal SEL. Further, the predetermined pattern can be longer or shorter than five values. Increasing the length of the pattern reduces the chance of a device being unintentionally unlocked.

If COM driver 220 sends a sequence of five data bytes matching the predefined pattern, counter 335 increments to final state and bit SEL2 of signal SEL[2:0] is set when a fifth byte is sent. With bit

SEL2 set, AND gate 333 asserting signal ADX causes flip-flops 337 and 338 to latch and store bits ADDR8 and ADDR4 of address signal ADDR[11:0] and causes flip-flop 345 to assert signal PCSYNC. Signal PCSYNC

5 indicates that the device is unlocked and has a base address of the form 001a 111b 1000, where signals PCA8 and PCA4 from flip-flops 337 and 338 indicate the values of bits a and b. Values of bits a and b have four possible combinations which allows COM driver 220

10 to select a combination that provides a base address that differs from the base addresses of the three other COM ports.

Signals PCA4 and PCA8 are latched when an address signal ADDR[11:0] is asserted for a byte following the predefined pattern. The byte following the predefined pattern does not match signal PAT[7:0] from pattern generator 310. Accordingly, counter 335 is reset to the initial state, and bit SEL2 is cleared. Signals PCA8 and PCA4 do not change unless the predefined

20 pattern is retransmitted. Unintentional transmission of the predefined pattern is unlikely during normal operation of the computer system, but if desired, COM driver 220 monitor the pattern being transmitted and prevent repetition of the predefined pattern, for

25 example by writing a no-op value to device 210.

Once the device is unlocked, a signal ADBASE indicates whether address signal ADDR[11:0] corresponds to the device. An AND gate 332 asserts a signal ADB if address signal ADDR[11:0] has the form 001x 111x 1xxx

30 when signal AEN indicates the address signal ADDR[11:0] is not from the DMA controller. A comparator 344 asserts signal ADBASE only if signal ADB is asserted, signal PCSYNC is asserted, and bits ADDR8 and ADDR4 of address signal ADDR[11:0] equal signals PCA8 and PCA4.

35 Conventional address decoding circuits (not shown) decode bits ADDR2, ADDR1, and ADDR0 to determine which

register in the device is being accessed via ISA bus 115.

The additional decoding circuits and the register set of the device can be implemented as required for the function of the device. The standard UART interface need not be followed. This allows an I/O interface to be optimized and implemented for the particular function of the device.

Fig. 2 shows an embodiment where serial device 210 contains an analog-to-digital converter (ADC) 206 and a digital-to-analog converter (DAC) 207 which are connected to PSTN phonelines 208 for implementation of a software modem. In this embodiment, software UART 222 and I/O handler 224 are part of a software modem 223. A register set in non-standard I/O interface 205 is described in Table 2.

Table 2

Offset	Register
0	Data Register (Low Byte)
1	Data Register (High Byte)
2	Control/Status Register (Low Byte)
3	Control/Status Register (High Byte)
4	Input/Output Port Register
5	Reserved
6	Reserved
7	Pattern Port Register

In the register set of Table 2, the data registers are for 16-bit data words sent to DAC 207 or received from ADC 206 by the host computer. Input/output port

register are for modem functions such as ring detection and control of an on-off hook relay (to connect or disconnect device 208 to an active phone line) which are implemented by hardware in serial device 210. The control/status register are general purpose control and status bit for serial device 210.

ADC 206 receives an analog communications signal from phonelines 208 and converts the analog communications signal into a series of sampled digital values. Software modem 223 receives the sampled digital values and based on the waveform represented by the sampled values and on the modem protocol employed determines data received. Software modem 223 also generates a series of digital values which are sent to DAC 207 and transmitted as an analog signal on phonelines 208. The transmitted analog signal provides a carrier signal and data values formatted according to standard modem protocols such as ITU V.32bis, V.32, V.22bis, V.23, V.22, V.21, V.17, V.29, and V.27ter standards. Device 210 generates periodic interrupts during which software modem 223 reads a set of sampled digital values from ADC 206 and writes a set of digital values which represent the transmitted analog signal. COM driver 220 sets the interrupt number (or IRQ) used by device 210 to a user selected one of eight values.

Application 140 communicates with software modem 223 in the same manner as with a conventional hardware modem. Application 140 sends and receives data and control values via operating environment 130. The data and control values are formatted for a standard UART device so that whether software modem 223 is a standard modem containing a hardware UART or a software modem is completely transparent to application 140 and operating environment 130.

Although the present invention has been described with reference to particular embodiments, the

description is only an example of the invention's
application and should not be taken as a limitation.
Various adaptations and combinations of features of the
embodiments disclosed will be apparent to those skilled
5 in the art and are within the scope of the present
invention as defined by the following claims.

05030710.002590

I claim:

1. A system comprising:

a computer having a processing unit, a main memory, and a local bus;

5 a device coupled to the local bus, wherein the device occupies an I/O slot which corresponds to a first communications port on the local bus, and the device has a register set which differs from a register set for a UART;

10 an operating system executed by the processing unit, wherein the operating system includes a procedure for accessing a register of a UART corresponding to the first communications port; and

15 a communications driver executed by the processing unit, comprising:

a UART emulation which in response to execution of the procedure for accessing a register of a UART, accesses storage locations in the main memory; and

20 an I/O handler which transfers values between the storage locations in the main memory and the register set of the device.

25 2. The system of claim 1, wherein the local bus comprises an ISA bus.

3. The system of claim 1, wherein the device coupled to the local bus, further comprises:

30 a comparator adapted for receiving a data signal from the local bus;

a pattern generator coupled to the comparator, wherein the pattern generator generates a signal for comparison with the data signal;

35

5 a counter operably coupled to the comparator,
wherein the counter resets to an initial state
following the comparator indicating the data
signal is not equal to the pattern signal and
advances toward a final state following the
comparator indicating the data signal equals the
pattern signal; and

10 a register coupled to the counter and adapted
to receive a signal from the local bus, wherein in
response to the counter reaching the final state,
the register latches from the local bus a value
which indicates the base address of the I/O slot
occupied by the device.

15 4. A method for communication between a computer
and a device having an I/O interface which differs from
the I/O interface of a UART, comprising:

coupling the I/O interface of the device to a
local bus in the computer;

20 allocating in a memory of the computer,
storage locations which correspond to registers of
a UART; and

transmitting values via the local bus between
the I/O interface of the device and the storage
25 locations in the memory of the computer.

5. The method of claim 4, further comprising
transmitting from an application to a communications
driver a data packet which is formatted for a UART,
30 wherein the communication driver updates a value in the
storage locations according to a value in the data
packet.

6. The method of claim 5, wherein the
35 communication drive performs the step of transmitting
by:

converting a value in the data packet to a converted value compatible with the I/O interface of the device; and

5 writing the converted value to a register in the device via the local bus.

7. The method of claim 4, wherein transmitting further comprising:

10 reading values from a register of the device via the local bus; and
updating the storage locations according to the value read.

8. The method of claim 7, further comprising
15 transmitting from a communications driver to an application a data packet containing a value from the storage locations.

9. The method of claim 4, further comprising:
20 executing on the computer an operating environment which allocates I/O slots on the local bus for UARTs; and
setting a base device address for the device to correspond to one of the I/O slots allocated by
25 the operating environment for a UART.

10. The method of claim 9, wherein setting the base device address comprises:
30 sensing, by the device, of a data signal on the local bus;
comparing by the data signal to a signal from a pattern generator in the device;
advancing a state indicator toward a final state in response to the data signal being equal
35 to the signal from the pattern generator;

repeating the steps of sensing, comparing, and advancing until the state indicator reaches the final state; and

5 setting the base address of the device to a value indicated by a signal on the local bus in response to the state indicator reaching the final state.

11. A device for connection to a local bus of a
10 computer, comprising:

 a comparator adapted for receiving a data signal from the local bus;

 a pattern generator coupled to the comparator, wherein the pattern generator
15 generates a signal for comparison with the data signal;

 a counter operably coupled to the comparator, wherein the counter resets to an initial state following the comparator indicating the data
20 signal is not equal to the pattern signal and advances toward a final state following the comparator indicating the data signal equals the pattern signal; and

 a register coupled to the counter and adapted
25 to receive a signal from the local bus, wherein in response to the counter reaching the final state, the register latches from the local bus a value which indicates the base address of the device.

30 12. The device of claim 11, further comprising an address decoder adapted for receiving an address signal from a local bus of a computer, wherein the address decoder selects which data signals the comparator receives from the local bus.

35

13. The device of claim 11, wherein the register is adapted to latch a signal from an address line of the local bus.

5 14. The device of claim 11, wherein the local bus comprises an ISA bus.

15 15. A method for setting the base address of a device coupled to a local bus of a computer,
10 comprising:
 sensing, by the device, of a data signal on the local bus;
 comparing by the data signal to a signal from a pattern generator in the device;
15 advancing a state indicator toward a final state in response to the data signal being equal to the signal from the pattern generator;
 repeating the steps of sensing, comparing, and advancing until the state indicator reaches
20 the final state; and
 setting the base address of the device to a value indicated by a signal on the local bus in response to the state indicator reaching the final state.

25 16. The method of claim 15, further comprising:
 placing the device in a locked state before the first sensing step, wherein in the locked state, the device does not respond to any signals
30 sensed on the local bus; and
 placing the device in an unlocked state after the step of setting the base address, wherein in the unlocked state the device responds to signals on the local bus which correspond to the set base
35 address of the device.

**COMMUNICATIONS INTERFACE AND CONFLICT AVOIDANCE
USING A SOFTWARE SIMULATION OF A UART**

Peter C. Chen

5 ABSTRACT OF THE DISCLOSURE

 A computer system includes a software UART emulation and uses standard operating system protocols to minimized the chance of I/O conflicts between a serial device having a UART and a non-standard serial device which communicates through the UART emulation. A COM driver which can replace a standard COM driver in an operating environment such as provided by Microsoft WINDOWSTM contains the UART emulation. The COM driver also sets the device address of the non-standard device on an ISA bus by determining the device addresses of COM port I/O slots used by UARTs, sending a predetermined pattern on the ISA bus to indicate a device address is to come, and then sending a value indicating a device address not used by the UARTs. The pattern has a length sufficient to make inadvertent generation of the pattern unlikely. The non-standard device recognizes the predetermined pattern and selects a device address according to the value from the COM driver.

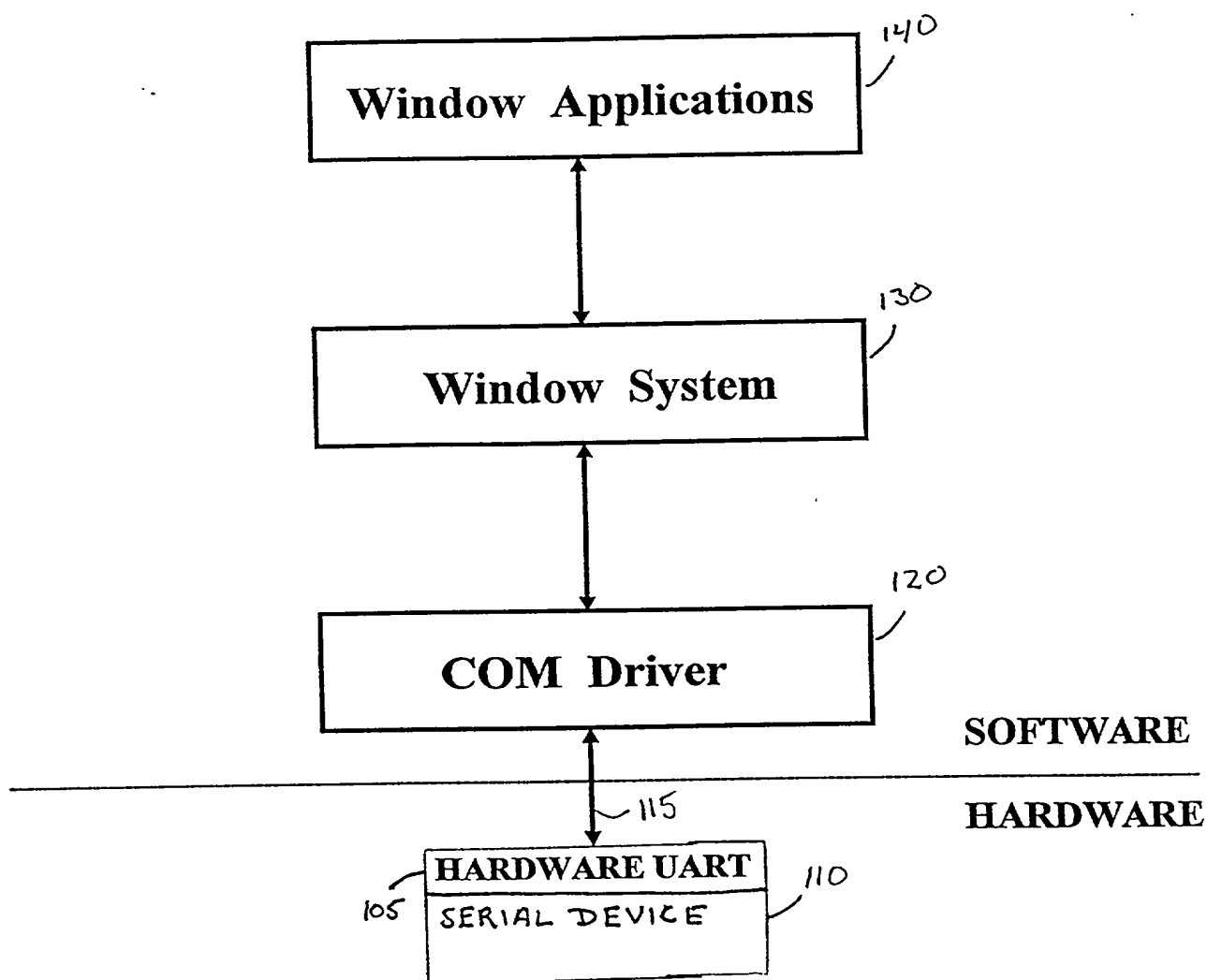


FIG. 1

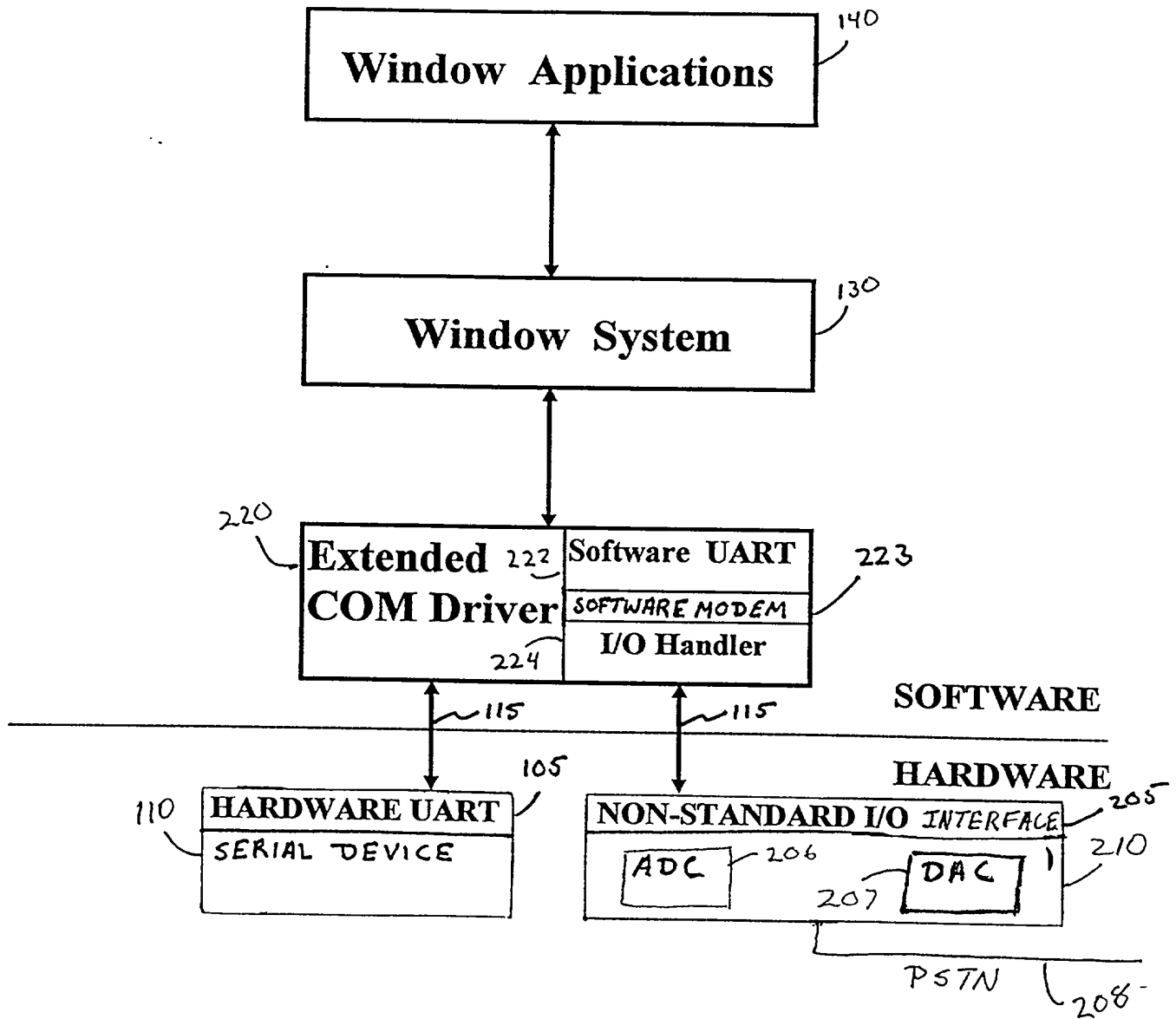


FIG. 2

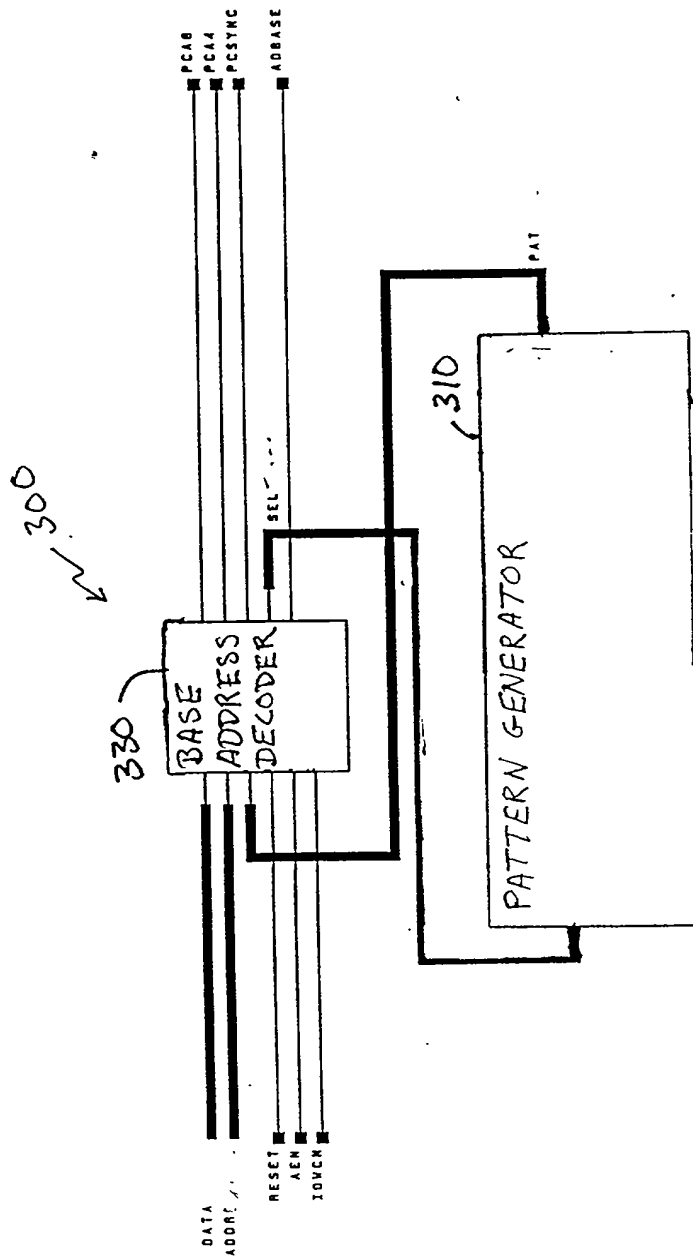


FIG. 3A

330

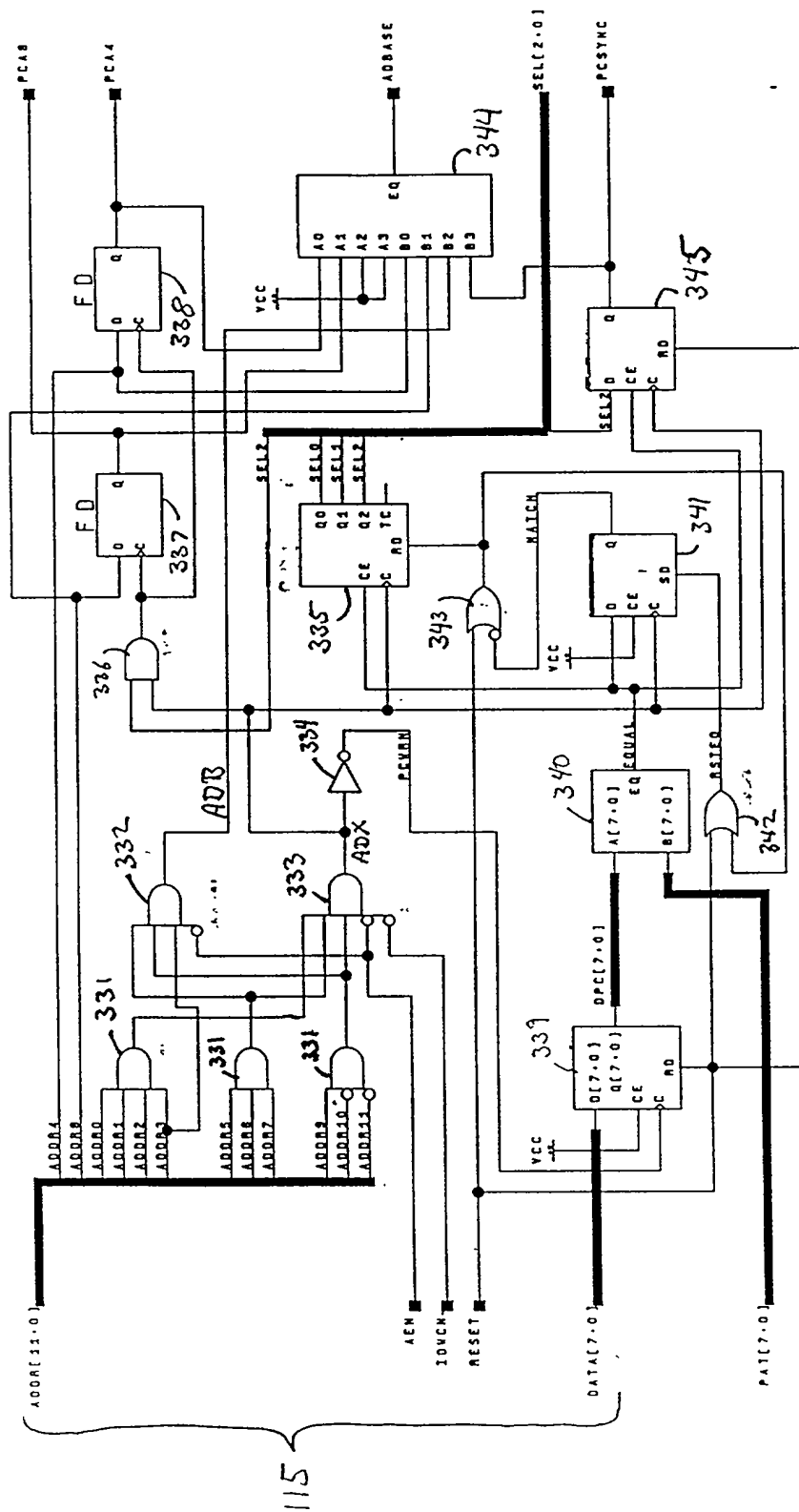


FIG. 3B

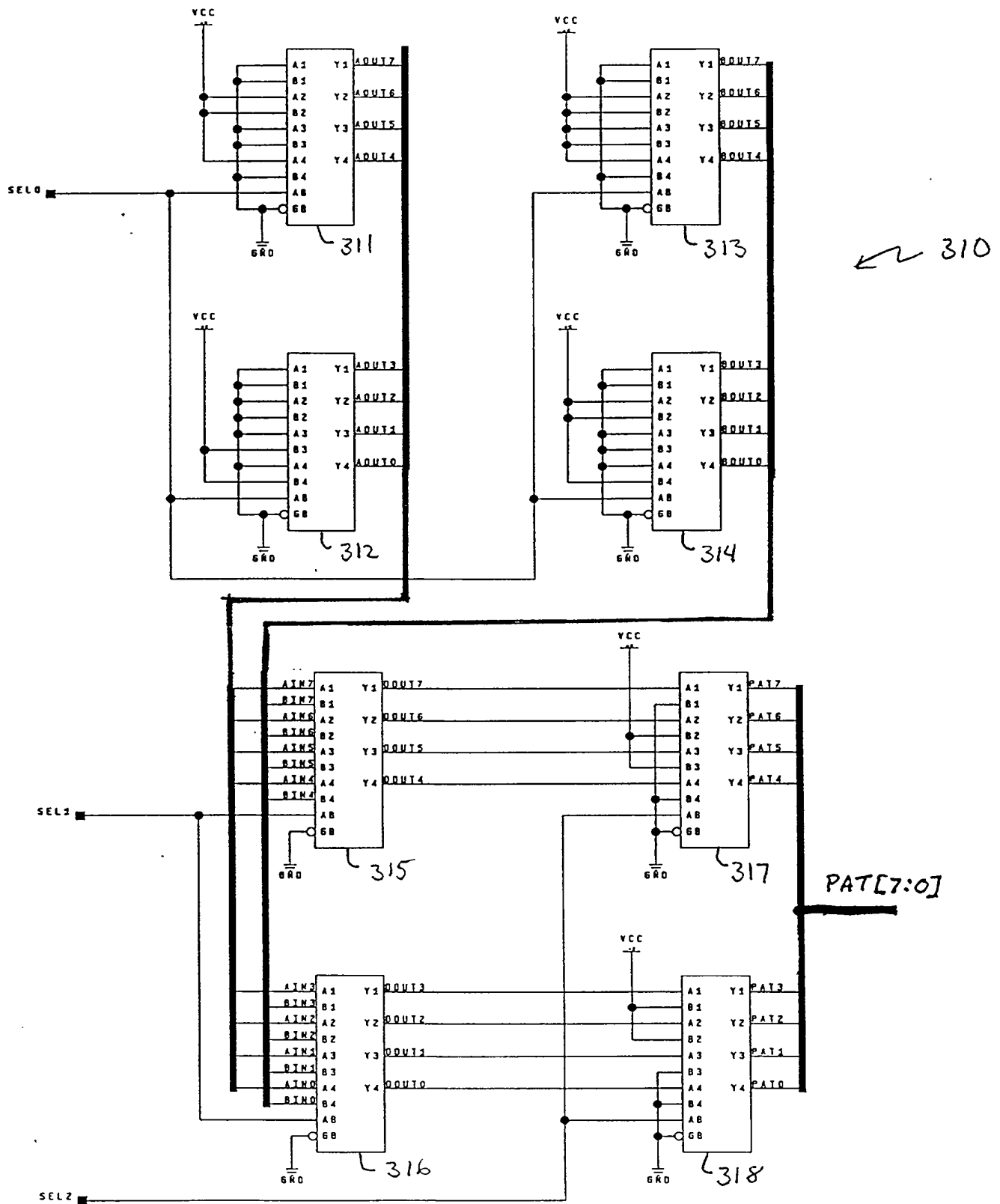


FIG. 3C

I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and to transact all business in the United States Patent and Trademark Office connected therewith:

Alan H. MacPherson (24,423); Thomas S. MacDonald (17,774); Richard Franklin (19,128); Kenneth E. Leeds (30,566); Paul J. Winters (25,246); Brian D. Ogonowsky (31,988); David W. Heid (25,875); Guy W. Shoup (26,805); Forrest E. Gunnison (32,899); Norman R. Klivans (33,003); David H. Carroll (29,903); Edward C. Kwok (33,938); Patrick T. Bever (33,834); David E. Steuber (25,557); Michael Shenker (34,250); Laura Terlizzi (31,307); T. Lester Wallace (34,748); Ronald J. Meetin (29,089); Andrew C. Graham (36,531); Ken John Koestner (33,004); Mark P. Kahler (29,178); Stephen A. Terrile (32,946); Omkar K. Suryadevara (36,320); David T. Millers (37,396); E. Eric Hoffman (38,186); and Lawrence E. Lycke (38,540).

Address all correspondence and telephone calls to:

ALAN H. MACPHERSON
Attorney for Applicant(s)
SKJERVEN, MORRILL, MacPHERSON, FRANKLIN & FRIEL
25 Metro Drive, Suite 700
San Jose, California 95110-1349

Telephone: 408-453-9200
Facsimile: 408-453-7979

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Title 18, United States Code, § 1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full name of sole or first inventor Peter C. Chen
Inventor's signature *Peter Chen* Date 4-25-95
Residence Saratoga, California Citizenship U.S.A.
Post Office Address 19849 Via Escuela Drive
Saratoga, CA 95070

Full name of second joint inventor, if any _____
Inventor's signature _____ Date _____
Residence _____ Citizenship _____
Post Office Address _____

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Peter C. Chen
 Assignee: Pctel, Inc.
 Title: "COMMUNICATIONS INTERFACE AND CONFLICT AVOIDANCE
 USING A SOFTWARE SIMULATION OF A UART"
 Serial No.: unknown Filed: Herewith
 Attorney Docket No.: M-3206 US

COMMISSIONER OF PATENTS AND TRADEMARKS
 Washington, D. C. 20231

SMALL BUSINESS CONCERN
 DECLARATION BY ASSIGNEE

Sir:

I declare that I am an official empowered to act on behalf of the concern identified above as assignee.

Exclusive rights to the above invention as described in
 [X] the specification filed herewith.

have been conveyed to and remain with the above concern.

For purposes of paying reduced fees under Section 41 of Title 35 of the United States Code with regard to this invention, I declare that the above concern qualifies as a small business concern as defined in 13 CFR 121.12 and reproduced in 37 CFR 1.9(d), namely, the concern's number of employees, including those of its affiliates, does not exceed 500 persons and the concern has not assigned, granted, conveyed, or licensed, and is under no obligation under contract or law to assign, grant, convey or license, any rights in the invention to any person who could not be classified as an independent inventor under 37 CFR 1.9(c) if that person made the invention, or to any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or a nonprofit organization under 37 CFR 1.9(e).

I acknowledge my duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate per 37 CFR 1.28(b).

I further declare that all statements made herein of my own knowledge are true, that all statements made on information and belief are believed to be true, that these statements are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of this application, or any patent issuing thereon.

Signature: Peter C. Chen Date: April 25, 1995
 Official's Name: Peter C. Chen
 Official's Title: President
 Concern's Name: Pctel, Inc.
 Concern's Address: 2680 N. First St., #210
San Jose, CA 95134

I hereby certify that this correspondence is being deposited with the United States Postal Service as express mail in an envelope addressed to: Commissioner of Patents and Trademarks, Washington, D.C., 20231, on April 25, 1995.
 Express Mail Receipt No. TB 856 096 22X US

4-25-95
 Date of Signature